

NAMA : RYU IRFAREYHAN AGAM
KELAS : 2IA11
NPM : 51422489

RANGKUMAN M1-M6

Pengenalan PBO

- ❖ Pemrograman berorientasi objek (Object Oriented Programming atau disingkat OOP) adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Objek adalah entitas yang memiliki atribut, karakter (bahaviour) dan kadang kala disertai kondisi (state)
- ❖ Pemrograman berorientasi objek ditemukan pada Tahun 1960, dimana berawal dari suatu pembuatan program yang terstruktur (structured programming). Metode ini dikembangkan dari bahasa C dan Pascal. Dengan program yang terstruktur inilah untuk pertama kalinya kita mampu menulis program yang begitu sulit dengan lebih mudah
- ❖ Pemrograman berorientasi objek dalam melakukan pemecahan suatu masalah tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut
- ❖ Pemrograman berorientasi objek bekerja dengan baik ketika dibarengi dengan Objek-Oriented Analysis And Design Process (OOAD).

Perbandingan PBO dan Terstruktur

PBO	TERSTRUKTUR
Menggabungkan fungsi dan data dalam kelas – kelas atau objek – objek	Memecah program dalam fungsi dan data
Memiliki ciri Encapsulation (pengemasan), Inheritance (penurunan sifat) dan Polymorphism (perbedaan bentuk dan perilaku)	Memiliki ciri Sequence (berurutan), Selection (pemilihan) dan Repetition (perulangan)
Struktur program ringkas, cukup dengan membuat Objek dan class lalu bekerja berdasarkan object dan class tersebut.	Struktur program rumit karena berupa urutan proses dan fungsi-fungsi
Kode program sangat re-usable. object dan class dapat digunakan berkali-kali, sehingga dapat menghemat space memori.	Re-use kode program kurang

Class pada PBO

Class adalah cetak biru atau blueprint dari object. Class digunakan hanya untuk membuat kerangka dasar.

CONTOH:

```
// Deklarasi Class
public class Kendaraan{
    // Ini Adalah Class, semua konstruktor, variabel, method berada disini
}
```

Objek pada PBO

Objek adalah entitas yang memiliki atribut, karakter dan kadang kala disertai kondisi. Objek mempresentasikan sesuai kenyataan seperti siswa, mempresentasikan dalam bentuk konsep seperti merek dagang, juga bisa menyatakan visualisasi seperti bentuk huruf (font).

CONTOH:

```
public class Kendaraan{

    // Konstruktor Dengan Parameter
    public Kendaraan(String nama){
        System.out.println("Nama Kendaraannya Adalah "+ nama);
    }

    public static void main(String[]args){

        // Perintah untuk membuat objek jenis
        Kendaraan jenis = new Kendaraan("Pesawat Terbang");

    }

}

// Output = Nama Kendaraannya Adalah Pesawat Terbang
```

Inheritance

Pewarisan merupakan sebuah bentuk “penggunaan kembali” (reusability); dimana class baru dibuat dari class yang pernah ada yang (biasanya) ditambah fasilitasnya. Setiap class turunan dapat menjadi class pokok (induk) untuk class turunan yang akan datang. Dalam pewarisan, constructor tidak diwariskan pada class turunannya, kecuali jika digunakan perintah super. Class yang mewariskan disebut dengan **superclass / parent class / base class**, sedangkan class yang mewarisi (class yang baru) disebut dengan **subclass / child class / derived class**. Untuk menerapkan inheritance, gunakan statement “**extends**”.

CONTOH:

Contoh untuk memanggil atribut dan method milik superclass-nya :

```
super.namaAtribut  
super.namaMethod(parameter)
```

Contoh untuk memanggil constructor milik superclass-nya :

```
super()  
super(parameter)
```

Encapsulation

Enkapsulasi (encapsulation) merupakan cara untuk melindungi property (atribut) / method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program. Cara untuk melindungi data yaitu dengan menggunakan access modifiers (hak akses). Ada 4 hak akses yang tersedia, yaitu default, public, protected, private.

No	Modifier	Pada class dan interface	Pada method dan variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method-method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method-method yang sepaket
4	private	Tidak bisa diterapkan	Tidak dapat diakses dimanapun kecuali oleh method-method yang ada dalam kelas itu sendiri

Polymorphism

Overloading adalah diperbolehkannya dalam sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda.

CONTOH:

```
1 public class Overloading {
2     public void Tampil(){
3         System.out.println("I love Java");
4     }
5     public void Tampil(int i){
6         System.out.println("Method dengan 1 parameter = "+i);
7     }
8     public void Tampil(int i, int j){
9         System.out.println("Method dengan 2 parameter = "+i+" & "+j);
10    }
11    public void Tampil(String str){
12        System.out.println(str);
13    }
14
15    public static void main(String a[]){
16        Overloading objek = new Overloading();
17        objek.Tampil();
18        objek.Tampil(8);
19        objek.Tampil(6,7);
20        objek.Tampil("Hello world");
21    }
22 }
```

Overriding method adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk (mendefinisikan kembali) method superclass-nya. Contoh overriding method dapat dilihat pada subclass “Mobil” yang mendefinisikan kembali method keterangan() dan hapus() dari class “Kendaraan”.